Descent Directions and Algorithmic Foundations

Optimization of Complex Systems – February 27th 2026

Andrea Brilli

Sapienza University of Rome

Minimizing a function

how does a solution of

$$\min_{x \in \mathbb{R}^n} f(x)$$

look like?

Key insight

A solution x^* is a point such that, **wherever we look around it**, the function can only go up (or stay flat).

Consequence: If we are NOT at a solution, there must exist a direction along which the function decreases!

Descent directions: the formal definition

Definition (Descent direction)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be defined on an open neighborhood of $x \in \mathbb{R}^n$. A direction $d \in \mathbb{R}^n$, $d \neq 0$ is a **descent direction** *for f at x when there exists $\bar{\alpha} > 0$ such that*

$$f(x + \alpha d) < f(x), \qquad \forall \alpha \in (0, \bar{\alpha}]$$

Question: If x^* is a local minimum, can a descent direction exist at x^*?

Answer: **NO!** If such a direction existed, we could move to a better point, contradicting optimality.

First-order characterization

But how do we *check* whether a direction is a descent direction?

Theorem (First-order descent condition)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable in a neighborhood of x, and let $d \in \mathbb{R}^n$, $d \neq 0$. If

$$\nabla f(x)^\top d < 0$$

then d is a descent direction for f at x.

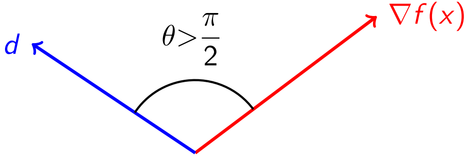
Proof idea: Consider the directional derivative

$$\lim_{t \to 0^+} \frac{f(x + td) - f(x)}{t} = \nabla f(x)^\top d < 0$$

For small t, the numerator must be negative, i.e. $f(x + td) < f(x)$.

Geometric interpretation

Recall: $\nabla f(x)^\top d = \|\nabla f(x)\| \, \|d\| \cos\theta$, where θ is the angle between $\nabla f(x)$ and d.



Condition: $\nabla f(x)^\top d < 0 \;\Leftrightarrow\; \cos\theta < 0 \;\Leftrightarrow\; \theta > \dfrac{\pi}{2}$

Best descent direction? $d = -\nabla f(x)$ (steepest descent) gives $\theta = \pi$.

The black-box setting

Problem: If we cannot compute $\nabla f(x)$, how do we find a descent direction?

Solution strategy: Use *multiple* search directions at each point!

Question: How should we choose these directions?

Positive spanning sets

Definition (Positive spanning set)

A finite set $D = \{d_1, \ldots, d_p\} \subset \mathbb{R}^n$ *is a* **positive spanning set** *if every vector* $v \in \mathbb{R}^n$ *can be written as a non-negative combination of vectors in* D*:*

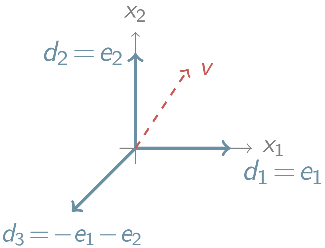
$$\forall\, v \in \mathbb{R}^n, \quad \exists\, \lambda_i \geq 0 : \quad v = \sum_{i=1}^{p} \lambda_i d_i$$

Key property: If D is a positive spanning set and $\nabla f(x) \neq 0$, then $\exists\, d_i \in D$ such that $\nabla f(x)^\top d_i < 0$.

Why? Take $v = -\nabla f(x)$. Then $\nabla f(x)^\top v < 0$, and $v = \sum \lambda_i d_i$ with $\lambda_i \geq 0$: at least one term must contribute negatively!

Example in $\mathbb{R}^2$

In $\mathbb{R}^2$, the simplest positive spanning set needs *three* directions:



Can we write any $v \in \mathbb{R}^2$ as

$$v = \lambda_1 d_1 + \lambda_2 d_2 + \lambda_3 d_3, \quad \lambda_i \geq 0\,?$$

Yes! For example:

$$v = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix} = 1 \cdot e_1 + 1.5 \cdot e_2 + 0 \cdot (-e_1 - e_2)$$

How much can we decrease?

We know:

- If $\nabla f(x)^\top d < 0$, then d is a descent direction
- For small α, we have $f(x + \alpha d) < f(x)$

Open questions:

1. How *much* can we decrease the function?
2. How does the decrease depend on the step size α?
3. What conditions on f guarantee predictable behavior?

Answer: We need to control how fast f and ∇f can change!

Lipschitz continuity

Definition (Lipschitz continuous function)

A function $f : \mathbb{R}^n \to \mathbb{R}$ is **Lipschitz continuous** *on $S \subseteq \mathbb{R}^n$ if there exists a constant $L \geq 0$ such that*

$$|f(x) - f(y)| \leq L\|x - y\|, \qquad \forall x, y \in S$$

The constant L is called the **Lipschitz constant**.

Interpretation: The function cannot change too rapidly – the rate of change is everywhere bounded by L.

Example

$f(x) = x^2$ on $[-1, 1]$ is Lipschitz continuous with $L = 2$.

Lipschitz continuous gradient

For optimization, the following condition on the gradient is even more useful:

Definition (Lipschitz continuous gradient)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. We say ∇f is **Lipschitz continuous** *on $S \subseteq \mathbb{R}^n$ with constant $L \geq 0$ if*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \qquad \forall x, y \in S$$

- **Meaning**: The gradient (slope) cannot change too abruptly – no sharp corners, no wild oscillations.
- **In algorithms**: This allows us to *predict* how the function behaves along a direction and choose step sizes accordingly.

Why do we care?

Intuition: If ∇f is Lipschitz continuous with constant L, then:

- The gradient at $x + \alpha d$ is "close" to the gradient at x
- We can approximate $f(x + \alpha d)$ using information at x
- We can bound the approximation error in terms of L and α

This leads to a fundamental result for optimization algorithms...

The descent lemma

Theorem (Descent Lemma)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable with Lipschitz continuous gradient with constant $L > 0$. Then, for all $x, y \in \mathbb{R}^n$:

$$f(y) \;\le\; f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2$$

Setting $y = x + \alpha d$:

$$f(x + \alpha d) \;\le\; f(x) + \alpha \nabla f(x)^\top d + \frac{L\alpha^2}{2}\|d\|^2$$

Two competing terms:

- Linear: $\alpha \nabla f(x)^\top d$ (first-order approximation)
- Quadratic: $\frac{L\alpha^2}{2}\|d\|^2$ (curvature penalty)

Understanding the descent lemma

Consider $y = x + \alpha d$ with $\nabla f(x)^\top d < 0$ (descent direction):

$$f(x + \alpha d) \;\leq\; f(x) + \alpha \underbrace{\nabla f(x)^\top d}_{<0} + \frac{L\alpha^2}{2}\|d\|^2$$

- **Linear term** $\alpha \nabla f(x)^\top d$: *negative* – drives decrease
- **Quadratic term** $\frac{L\alpha^2}{2}\|d\|^2$: *positive* – limits decrease for large α

Key insight: For α small enough, the linear term dominates!

Why is this useful for algorithms?

The descent lemma gives us:

1. **Guaranteed decrease**: If $\nabla f(x)^\top d < 0$, choosing α small enough ensures $f(x + \alpha d) < f(x)$.
2. **Convergence analysis**: Algorithms that enforce this decrease can be shown to converge to stationary points.
3. **Black-box algorithms**: Even without ∇f, a positive spanning set guarantees that at least one direction will yield descent.

The algorithmic picture

Goal: Solve $\min_{x \in \mathbb{R}^n} f(x)$.

What we now know:

- If we are not at a solution, descent directions exist
- A direction d is descent if $\nabla f(x)^\top d < 0$
- If ∇f is unavailable, a positive spanning set suffices
- The descent lemma tells us how much we can decrease, and how

Gradient descent template:

1. Choose $x_0 \in \mathbb{R}^n$
2. While $\nabla f(x_k) \neq 0$:
   - Set $d_k = -\nabla f(x_k)$
   - Choose step size α_k (e.g. $\alpha_k = 1/L$)
   - Update $x_{k+1} = x_k + \alpha_k d_k$

For black-box functions

If ∇f is unavailable, use a **positive spanning set** $D = \{d_1, \ldots, d_p\}$:

1. Choose $x_0 \in \mathbb{R}^n$, initial step size $\alpha > 0$
2. At iteration k:
   - For each $d_i \in D$: try $x_k + \alpha d_i$
   - If $f(x_k + \alpha d_i) < f(x_k)$ for some i: set $x_{k+1} = x_k + \alpha d_i$ (*success*)
   - If no direction succeeds: reduce α and try again

Key guarantee: If $\nabla f(x_k) \neq 0$, at least one d_i in a positive spanning set is a descent direction.

Practical note: Step size reduction follows the descent lemma's guidance on how small α must be to ensure decrease.

Connection to first-order necessary conditions

Proposition (First-order necessary condition – recall)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. If x^* is a local minimum, then

$$\nabla f(x^*) = 0.$$

Contrapositive: $\nabla f(x) \neq 0 \;\Rightarrow\; x$ is NOT a local minimum.

Indeed, $d = -\nabla f(x)$ satisfies

$$\nabla f(x)^\top d = -\|\nabla f(x)\|^2 < 0,$$

so d is a descent direction.

Algorithmic philosophy: Keep moving along descent directions until $\|\nabla f(x_k)\| \approx 0$.

From theory to practice

What we know so far:

- If $\nabla f(x) \neq 0$, descent directions exist
- Positive spanning sets guarantee we can find descent directions
- The descent lemma quantifies how much decrease we can achieve

The practical challenge:

What if computing $\nabla f(x)$ is impossible or too expensive?

Question: How can we design an algorithm that uses only function values?

A historical perspective

1952: Fermi and Metropolis faced exactly this problem.

- Problem: Eastimate 6 unknown parameters (phase shifts) from experimental data
- Challenge: Function $M(\alpha)$ available only through computation ($\sim$0.4 seconds per evaluation)
- Computer: MANIAC I at Los Alamos — very limited memory and speed
- No gradient information available

The simplest positive spanning set

We proved that in $\mathbb{R}^2$, we need at least 3 directions for a positive spanning set.

Question: What if we're willing to use MORE directions? What's the most natural, symmetric choice in $\mathbb{R}^n$?

Answer: The **coordinate directions**!

$$D = \{\pm e_1, \pm e_2, \dots, \pm e_n\}$$

where e_i is the i-th canonical basis vector.

How many directions? $|D| = 2n$

Is this a positive spanning set? Let's verify!

The simplest positive spanning set

We proved that in $\mathbb{R}^2$, we need at least 3 directions for a positive spanning set.

Question: What if we're willing to use MORE directions? What's the most natural, symmetric choice in $\mathbb{R}^n$?

Answer: The **coordinate directions**!

$$D = \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$$

where e_i is the i-th canonical basis vector.

How many directions? $|D| = 2n$

Is this a positive spanning set? Let's verify!

Coordinate directions form a positive spanning set

Proposition

The set $D = \{\pm e_1, \ldots, \pm e_n\}$ is a positive spanning set for $\mathbb{R}^n$.

Proof: Let $v \in \mathbb{R}^n$ be arbitrary. Write $v = (v_1, v_2, \ldots, v_n)^\top$.

Define:

$$\lambda_i^+ = \begin{cases} v_i & \text{if } v_i \geq 0 \\ 0 & \text{if } v_i < 0 \end{cases}, \qquad \lambda_i^- = \begin{cases} 0 & \text{if } v_i \geq 0 \\ |v_i| & \text{if } v_i < 0 \end{cases}$$

Then $\lambda_i^+, \lambda_i^- \geq 0$ for all i, and:

$$v = \sum_{i=1}^{n} v_i e_i = \sum_{i=1}^{n} \lambda_i^+ e_i + \sum_{i=1}^{n} \lambda_i^- (-e_i)$$

This is a non-negative combination of vectors in D. $\square$

Why coordinate directions?

Advantages of using $D = \{\pm e_1, \ldots, \pm e_n\}$:

- **Simplicity**: Move along one coordinate axis at a time
- **Easy to implement**: Just add or subtract α from x_i
- **No complex calculations**: No need to compute or approximate gradients
- **Natural interpretation**: Variables often represent independent quantities
- **Minimal storage**: Only need to store current point and step size

Drawback: May require $2n$ function evaluations per iteration.

Coordinate search algorithm

Basic idea:

1. At current point x, try moving along each coordinate direction $\pm e_i$ with step size α
2. If any move improves f, accept it and continue with same α
3. If no move improves f, reduce step size: $\alpha \leftarrow \theta\alpha$ (with $\theta \in (0, 1)$)
4. Repeat until α becomes sufficiently small

Step size strategy:

- Success $\Rightarrow$ keep α (the current scale is working)
- Failure $\Rightarrow$ reduce α (search at finer scale)

Key parameter: $\theta \in (0, 1)$ controls step size reduction (typical:

Coordinate search: pseudocode

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, $\alpha_{\min} > 0$, $\theta \in (0,1)$

$k \leftarrow 0$, $x \leftarrow x_0$, $\alpha_k \leftarrow \alpha_0$
while $\alpha \geq \alpha_{\min}$ **do**
  $k \leftarrow k + 1$
  Find $\bar{d} \in D = \{\pm e_1, \dots, \pm e_n\}$ s.t. $f(x_k + \alpha_k \bar{d}) < f(x_k)$
  if such $\bar{d}$ exists **then**
    $x_{k+1} \leftarrow x_k + \alpha_k \bar{d}$, $\alpha_{k+1} \leftarrow \alpha_k$ (success: keep step size)
  else
    $\alpha_{k+1} \leftarrow \theta \alpha_k$ (failure: reduce step size)
  end if
end while
return x_k

Stopping criterion: $\alpha_k < \alpha_{\min}$ (step size becomes negligible)

Implementation variants

Opportunistic search

Find *any* improving direction and stop searching:

- Faster per iteration (fewer function evaluations)
- May accept small improvements and miss better directions

Best improvement search

Evaluate *all* $2n$ directions and choose the best:

$$\bar{d} \in \arg\min_{d \in D} f(x + \alpha d)$$

- $2n$ function evaluations per iteration
- Guarantees largest decrease among coordinate directions

Which to choose? Best improvement often more efficient overall despite higher per-iteration cost.

Example in $\mathbb{R}^2$: setup

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

- **Starting point**: $x_0 = (-0.9, -1.0)^\top$
- **Initial function value**: $f(x_0) = 11.3524$
- **Initial step size**: $\alpha_0 = 0.3$
- **Reduction factor**: $\theta = 0.5$
- **Directions**: $D = \{e_1, -e_1, e_2, -e_2\}$ (East, West, North, South)

We'll use the **best improvement** variant: evaluate all 4 directions and choose the best.

Compass Search

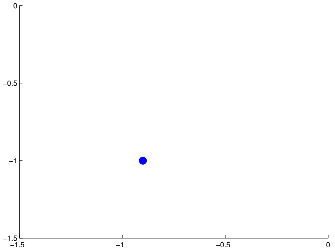
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Initial Point: $x_0 = (-0.9; -1.0)^\top$

$f(x)$ initial: $f(x_0) = 11.3524$

Initial step-size: $\Delta = 0.3$

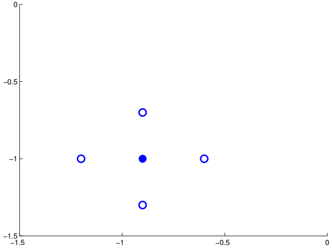


Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Initial Point: $x_0 = (-0.9; -1.0)^\top$
$f(x)$ initial: $f(x_0) = 11.3524$
Initial step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

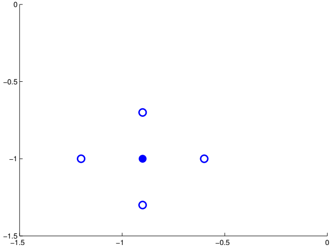
East	11.7904
West	19.9504
North	
South	29.4628

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Initial Point: $x_0 = (-0.9; -1.0)^\top$
$f(x)$ initial: $f(x_0) = 11.3524$
Initial step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

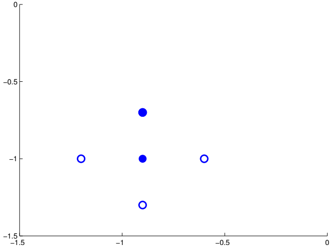
East	11.7904
West	19.9504
North	5.0788
South	29.4628

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.9; -0.7)^\top$
$f(x)$ current: $f(x_k) = 5.0788$
Current step-size: $\Delta = 0.3$



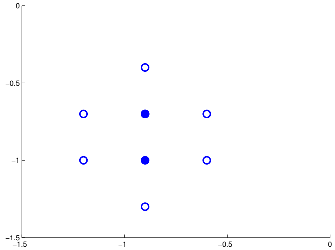
Let us compute $f(x)$ at the following candidates

East	11.7904
West	19.9504
North	**5.0788**
South	29.4628

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$



Current point: $x_k = (-0.9; -0.7)^\top$
$f(x)$ current: $f(x_k) = 5.0788$
Current step-size: $\Delta = 0.3$

Let us compute $f(x)$ at the following candidates

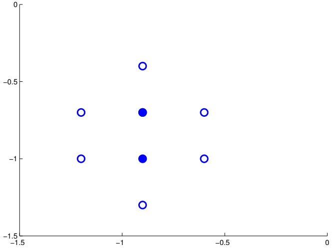
East	
West	17.4208
North	6.4948
South	11.3524

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.9; -0.7)^\top$
$f(x)$ current: $f(x_k) = 5.0788$
Current step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

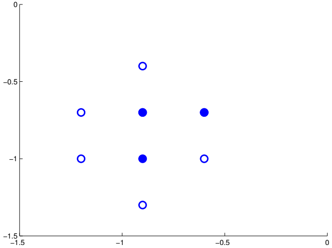
East	2.2048
West	17.4208
North	6.4948
South	11.3524

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.7)^\top$
$f(x)$ current: $f(x_k) = 2.2048$
Current step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

East	**2.2048**
West	17.4208
North	6.4948
South	11.3524

Compass Search

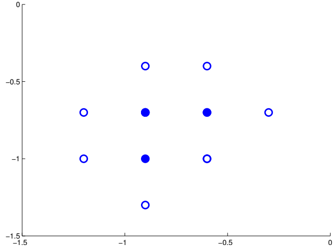
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.7)^\top$

$f(x)$ current: $f(x_k) = 2.2048$

Current step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

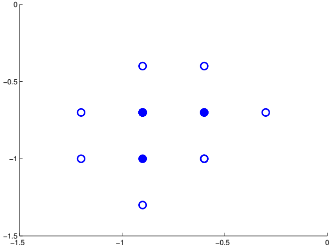
East	4.9108
West	5.0788
North	
South	11.7904

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.7)^\top$
$f(x)$ current: $f(x_k) = 2.2048$
Current step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

East	4.9108
West	5.0788
North	0.5248
South	11.7904

Compass Search

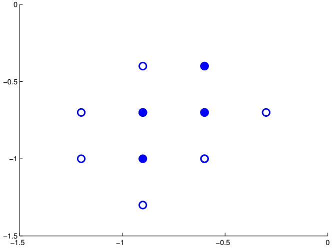
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.5248$

Current step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

East	4.9108
West	5.0788
North	**0.5248**
South	11.7904

Compass Search

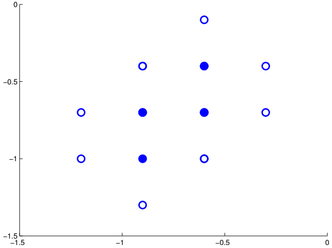
Consider the problem:

$$\min_{x\in\mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.5248$

Current step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

East	0.5668
West	6.4948
North	3.3808
South	2.2048

Compass Search

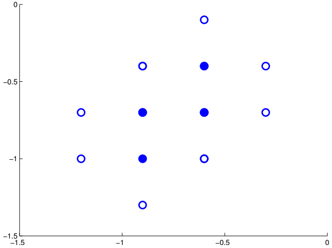
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.5248$

Current step-size: $\Delta = 0.3$



Let us compute $f(x)$ at the following candidates

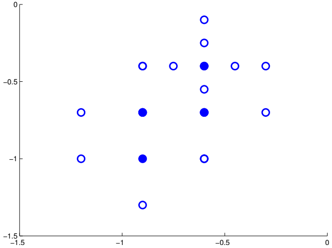
East	0.5668
West	6.4948
North	3.3808
South	2.2048

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.4)^\top$
$f(x)$ current: $f(x_k) = 0.5248$
Current step-size: $\Delta = 0.15$



Let us compute $f(x)$ at the following candidates

East	
West	2.5335
North	1.5660
South	0.6054

Compass Search

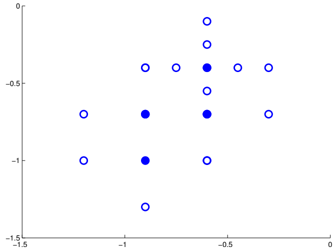
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.6; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.5248$

Current step-size: $\Delta = 0.15$



Let us compute $f(x)$ at the following candidates

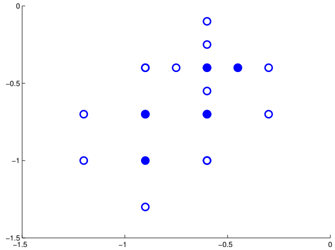
East	0.0069
West	2.5335
North	1.5660
South	0.6054

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.45; -0.4)^\top$
$f(x)$ current: $f(x_k) = 0.0069$
Current step-size: $\Delta = 0.15$



Let us compute $f(x)$ at the following candidates

East	**0.0069**
West	2.5335
North	1.5660
South	0.6054

Compass Search

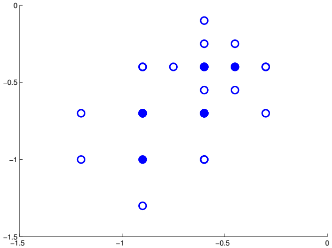
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.45; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.0069$

Current step-size: $\Delta = 0.15$



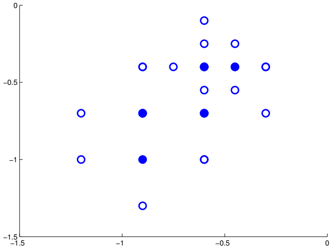
Let us compute $f(x)$ at the following candidates

East	0.5668
West	0.5248
North	0.3957
South	0.7670

Compass Search

Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$



Current point: $x_k = (-0.45; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.0069$

Current step-size: $\Delta = 0.15$

Let us compute $f(x)$ at the following candidates

East	0.5668
West	0.5248
North	0.3957
South	0.7670

Compass Search

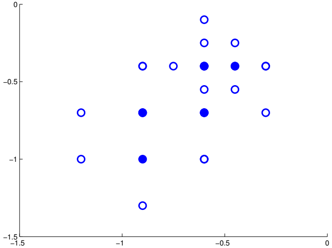
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.45; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.0069$

Current step-size: $\Delta = 0.075$



Let us compute $f(x)$ at the following candidates

East	0.5668
West	0.5248
North	0.3957
South	0.7670

Compass Search

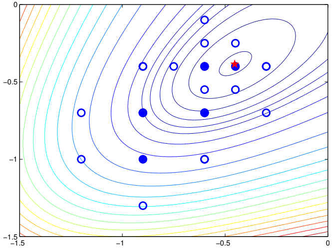
Consider the problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

Current point: $x_k = (-0.45; -0.4)^\top$

$f(x)$ current: $f(x_k) = 0.0069$

Current step-size: $\Delta = 0.075$



Let us compute $f(x)$ at the following candidates

East	0.5668
West	0.5248
North	0.3957
South	0.7670

Summary

$f(x_k)$	Δ_k
11.352400	0.300000
5.078800	0.300000
2.204800	0.300000
0.524800	0.300000
0.524800	0.150000
0.006925	0.150000
0.006925	0.075000
0.006925	0.037500
0.006925	0.018750
0.000298	0.018750
0.000298	0.009375
0.000298	0.004687
0.000298	0.002344
0.000173	0.002344

Convergence analysis: setup

Assumption:

(A1) The level set $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is **compact**.

Convergence analysis: setup

Assumption:

(A1) The level set $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is **compact**.

Key observations:

1. The sequence $\{f(x_k)\}$ is non-increasing: $f(x_{k+1}) \leq f(x_k)$
2. All iterates remain in $\mathcal{L}(x_0)$
3. Step size α_k either stays constant or decreases: $\alpha_{k+1} \in \{\alpha_k, \theta\alpha_k\}$